PLAN, GUIDE, AND REPAIR: DECOMPOSED CONTROL FOR CONDITIONAL MOLECULAR GRAPH DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Conditional molecular graph generation must align requested properties without losing the discrete chemical structure that makes a sample usable. Existing generators typically leave molecular size, property control along the reverse trajectory, and chemical correction to a single denoising process, although these decisions require different information at different stages. We introduce *Plan, Guide, and Repair* (PGR), a sampling framework that controls them separately while keeping the generative backbone frozen. Plan selects a condition-dependent, training-supported molecular size before denoising. Guide learns the endpoint property implied by sampled categorical states from actual reverse trajectories and, at every reverse step, differentiates the target discrepancy on soft node and edge distributions to steer generation toward the requested condition. Repair then uses the backbone edge posterior to revise late bond choices that violate valence, aromaticity, or connectivity constraints. On joint $O_2/N_2/CO_2$ generation, PGR attains 0.9991 raw validity and 0.497 average property error. On BACE, BBBP, and HIV, it reaches property accuracies of 0.9716, 0.9665, and 0.9912 with raw validities of 0.9248, 0.9549, and 0.7745. Ablations confirm the intended roles of all three components: Plan improves condition-aware size selection, Guide improves condition alignment by actively controlling reverse generation, and Repair restores raw chemical validity while preserving the alignment gain. These results support stage-specific sampling control across continuous and categorical molecular design tasks.

1 INTRODUCTION

Inverse molecular design asks a generator to construct a molecular graph whose properties match a requested specification. This capability is central to drug discovery and polymer membrane design, where candidates must satisfy numerical or categorical targets while remaining chemically meaningful (Du et al., 2024; Wu et al., 2018; Ma & Luo, 2020; Barnett et al., 2020; Robeson, 2008; Gebauer et al., 2022). Diffusion models are attractive for this problem because they turn sampling into a sequence of correctable decisions and, on graphs, operate directly on atom and bond variables (Ho et al., 2020; Song et al., 2021; Hoogeboom et al., 2022; Vignac et al., 2023). The difficulty is therefore not merely to produce a plausible graph, but to control a discrete stochastic trajectory so that its endpoint has the requested properties and remains chemically valid.

Recent conditional backbones have made substantial progress on this problem. Graph DiT adapts transformer conditioning to heterogeneous numerical and categorical targets and couples atom and bond corruption through graph-dependent noise (Vaswani et al., 2017; Liu et al., 2024). MELD builds on masked discrete diffusion and learns element-wise forward trajectories that reduce collisions between semantically different graphs (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024; Seo et al., 2026). These advances strengthen the learned generator, but three limitations remain at different stages of conditional generation. Molecular size is commonly inherited from an empirical prior rather than selected for the requested condition. Auxiliary property predictors may be trained on clean or artificially corrupted molecules even though guidance is applied to imperfect states produced by the reverse sampler. Finally, a property-aligned edge proposal can still be overvalent, assign aromaticity outside a cycle, or leave disconnected fragments. *Improving the backbone alone does not determine when or how these three decisions should be controlled.*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The decisions are coupled in the final molecule but differ in scale and timing. Molecular size is a global choice that fixes the active graph support before topology exists. Property alignment is a trajectory-level choice: the controller must estimate what the current reverse state is likely to become and redirect the sampler while uncertainty remains. Chemical consistency is dominated by local discrete choices after atom and bond identities have largely resolved. Folding all three roles into a single conditioning vector or property objective gives no explicit mechanism for matching size to the target, aligning predictor supervision with the reverse-state distribution, or preventing a locally probable bond from violating graph-level chemistry.

We introduce **Plan, Guide, and Repair (PGR)**, a sampling framework that places each decision at the stage where its information is available. Plan predicts a training-supported node count from the requested condition and fixes the resulting mask before reverse diffusion. Guide is supervised on sampled categorical states collected from actual planner-conditioned reverse trajectories and predicts the property of each trajectory's final generated endpoint. During generation, the same predictor f_ϕ is evaluated differentiably on p_X^0, p_E^0; the discrepancy from the requested target defines ℓ_t, whose gradient actively steers the reverse process. Repair operates only at selected late steps. It ranks alternative edge choices with the backbone posterior and accepts an edit only when it reduces explicit valence, aromaticity, or connectivity violations without changing node identities or molecular size. PGR therefore augments a pretrained conditional generator without retraining its backbone.

Our contribution is a unified decomposition of conditional graph sampling into three testable controls. **Plan** turns molecular size from a condition-agnostic draw into a supported conditional decision. **Guide** aligns its trajectory source and supervision target with the reverse-generation question it is used to answer, then converts that prediction into an all-step sampling signal. **Repair** introduces bounded, posterior-supported chemical correction inside the reverse chain rather than filtering or replacing completed molecules. We evaluate the complete method on joint $O_2/N_2/CO_2$ polymer generation and on BACE, BBBP, and HIV small-molecule generation. The main comparisons show that PGR improves condition alignment over the reported baselines while maintaining strong raw validity, and the controlled ablations isolate the intended contribution of each stage.

2 Related Work

Molecular generation and goal-directed optimization. Variational and flow-based generators amortize sampling in a learned molecular distribution, from general latent-variable and flow constructions to graph-specific models such as GraphVAE, JTVAE, GraphAF, and MoFlow (Kingma & Welling, 2014; Rezende & Mohamed, 2015; Simonovsky & Komodakis, 2018; Jin et al., 2018; Shi et al., 2020; Zang & Wang, 2020). Goal-directed alternatives optimize an oracle through graph mutation, reinforcement learning, recurrent sequences, Markov sampling, interpretable substructures, differentiable scaffolds, or conditional generation (Jensen, 2019; You et al., 2018; Segler et al., 2018; Sanchez-Lengeling & Aspuru-Guzik, 2018; Xie et al., 2021; Jin et al., 2020; Fu et al., 2022; Mollaysa et al., 2020). These methods are useful when a score can be queried repeatedly, but each requested vector can require a new search or a scalarization of heterogeneous targets. Benchmark studies further show that sample budget and apparently simple search baselines materially affect conclusions, and that molecular optimization has recognizable failure modes (Brown et al., 2019; Gao et al., 2022; Tripp & Hernández-Lobato, 2023; Renz et al., 2019). PGR instead keeps an amortized conditional generator and allocates sampling controls to graph size, trajectory consequence, and local chemistry.

Continuous, discrete, and masked graph diffusion. Diffusion probabilistic models and score-based SDEs established iterative denoising in continuous spaces (Ho et al., 2020; Nichol & Dhariwal, 2021; Song et al., 2021); categorical diffusion and continuous-time discrete formulations extend the principle to finite state spaces (Austin et al., 2021; Hoogeboom et al., 2021; Campbell et al., 2022). For graphs and molecules, score-based models generate permutation-invariant graphs or coupled node–edge variables, while categorical and equivariant variants address discrete topology or 3D geometry (Niu et al., 2020; Jo et al., 2022; Vignac et al., 2023; Hoogeboom et al., 2022). MOOD adds property and out-of-distribution guidance, Graph DiT adds heterogeneous multi-condition encoding and graph-dependent corruption, and masked models progressively infill hidden elements (Lee et al., 2023; Liu et al., 2024; Sahoo et al., 2024; Shi et al., 2024). MELD improves the masked corruption path itself by learning element-specific schedules (Seo et al., 2026). These advances chiefly change the learned distribution, denoiser, or forward process. They do not jointly decide

condition-dependent size, train a controller on states actually produced by reverse sampling, and enforce late bond constraints; PGR is designed around those remaining sampling decisions.

Conditional diffusion and property guidance. Guidance may enter through the denoiser, a classifier gradient, or an implicit conditional–unconditional score difference (Dhariwal & Nichol, 2021; Ho & Salimans, 2022). Molecular variants use energy functions or differentiable property models to steer inverse design (Bao et al., 2023; Weiss et al., 2023). Their success establishes the value of gradients, but does not remove a distributional question: a predictor trained on prescribed forward corruptions is asked at sampling time about imperfect states created by a learned reverse chain. Guide trains on those reverse states and labels each with its own eventual generated endpoint. The same predictor then supplies an all-step target-discrepancy gradient, so endpoint learning is the prospective signal for changing the generation path rather than a post hoc property estimate.

Chemical representation, validity, and evaluation. Molecular evaluation depends on representation and proxy choices: SMILES specifies graph serialization, extended-connectivity fingerprints support similarity calculations, SA and SCScore approximate synthesizability, and FCD compares generated and reference distributions in a learned chemical space (Weininger, 1988; Rogers & Hahn, 2010; Ertl & Schuffenhauer, 2009; Coley et al., 2018; Preuer et al., 2018). GuacaMol, MOSES, and MoleculeNet standardized complementary generation and prediction tests (Brown et al., 2019; Polykovskiy et al., 2020; Wu et al., 2018). Yet corrected validity may include conversion rules, component selection, or invalid-sample removal. Hard rejection discards computation, while unconstrained property guidance does not ensure that a high-scoring edge assignment respects valence, aromaticity, or connectivity. Repair instead acts inside the reverse chain, preserves nodes and graph size, restricts edits to posterior-supported bond classes, and accepts only monotonic reductions of explicit violations. Raw validity therefore measures the graph emitted by the sampler rather than the success of a downstream cleanup rule.

3 METHOD

3.1 CONDITIONAL GRAPH DIFFUSION WITH DECOMPOSED CONTROL

Let $Z_t = (X_t, E_t)$ denote a molecular graph state with categorical node variables X_t, symmetric categorical edge variables E_t, and requested condition $\mathbf{c}$. A pretrained conditional graph diffusion model defines

$$Z_T \to Z_{T-1} \to \cdots \to Z_0, \qquad p_\theta(Z_{t-1} \mid Z_t, \mathbf{c}), \tag{1}$$

where θ remains frozen. At reverse step t, the backbone's current predictions for the clean node and edge categories are denoted p_X^0 and p_E^0; their dependence on t is implicit. A hard state is then sampled from the resulting reverse transition. PGR adds three controls without changing the denoising objective or backbone parameters. Plan selects the active node mask before Eq. 1 begins. Guide modifies p_X^0 and p_E^0 using a differentiable estimate of the trajectory endpoint. Repair evaluates hard edge proposals at a small set of late steps. The decomposition follows the information revealed by sampling: the target is sufficient to plan graph size, an evolving soft prediction is needed to steer endpoint properties, and resolved bond proposals are needed to diagnose chemical violations. Figures 1–3 visualize these complementary interventions.

3.2 PLAN: CONDITION-AWARE MOLECULAR SIZE

Sampling a node count from an unconditional empirical prior ignores information already present in $\mathbf{c}$ and fixes the dimensionality of all later decisions. We instead learn a node-count predictor $q_\omega(N \mid \mathbf{c})$ on training pairs $(\mathbf{c}_i, N_i)$ with cross-entropy

$$\mathcal{L}_{\text{plan}}(\omega) = -\sum_i \log q_\omega(N_i \mid \mathbf{c}_i). \tag{2}$$

Let $\mathcal{N}_{\text{train}}$ denote the node counts observed in the training split. At inference, unsupported counts are assigned zero probability and the planned size is

$$\hat{N}(\mathbf{c}) = \arg\max_{N \in \mathcal{N}_{\text{train}}} q_\omega(N \mid \mathbf{c}). \tag{3}$$

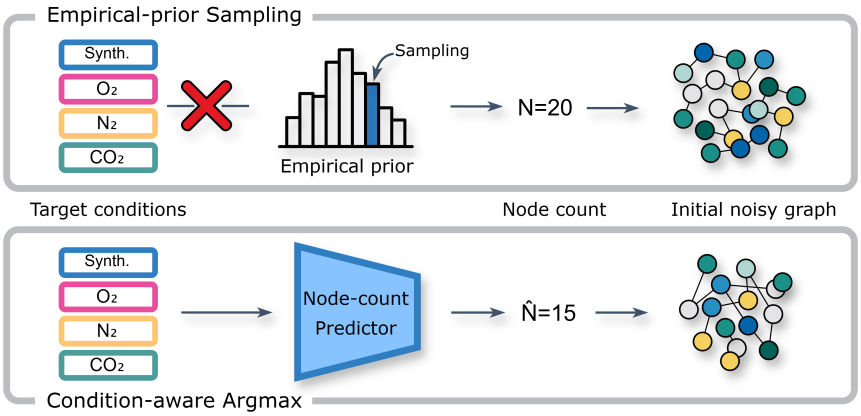

Figure 1: **Plan makes molecular size condition-aware before reverse diffusion.** Sampling from an empirical node-count prior ignores the requested properties. Plan instead applies the node-count predictor $q_\omega(N \mid \mathbf{c})$ and selects the training-supported $\hat{N}(\mathbf{c})$ before instantiating the initial noisy graph. The displayed node counts are illustrative.

The resulting mask $M(\mathbf{c})$ activates exactly $\hat{N}$ nodes and is held fixed throughout sampling. Numerical conditions use their training normalization and categorical conditions retain their categorical encoding. Plan therefore changes the global support of the sampled graph, but neither modifies the requested properties nor extrapolates to an unseen molecular size.

3.3 Guide supervision from reverse trajectories

Instead of constructing predictor inputs by applying prescribed forward corruptions to clean training graphs, we collect states actually sampled along the backbone's reverse trajectories. Unlike forward-corrupted states anchored to known clean molecules, these states reflect the backbone's own accumulated sampling errors and uncertainty. Pairing each state with its eventual generated endpoint therefore provides supervision that more closely matches the trajectory consequences Guide must control during generation. With the backbone frozen, Plan active, and Guide and Repair disabled, we sample complete trajectories using training-split conditions. At a retained timestep, the categorical node and edge state is the graph actually sampled by the reverse transition,

$$Z_t^{\text{hard}} \sim p_\theta(Z_t \mid Z_{t+1}, \mathbf{c}), \tag{4}$$

not the argmax of a stored soft posterior. We encode its sampled categories as one-hot node and edge features, retain the normalized time t/T, and pair it with the same trajectory's final generated endpoint Z_0. The supervision target $\mathbf{y}(Z_0)$ is the evaluated property of Z_0, not the input condition or the property of an unrelated clean molecule. The one-hot training inputs and the probability-valued guidance inputs use the same node and edge category axes.

Let $m(Z_0)$ be one for an evaluable endpoint and zero otherwise. For continuous target indices $\mathcal{J}$, Guide predicts normalized endpoint properties with

$$\mathcal{L}_{\text{cont}}(\phi) = \frac{m(Z_0)}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \left| f_{\phi,j}(Z_t^{\text{hard}}, t) - y_j(Z_0) \right|. \tag{5}$$

For a binary target, the trajectory endpoint label $y(Z_0)$ gives

$$\mathcal{L}_{\text{cat}}(\phi) = m(Z_0)\, \text{BCEWithLogits}\big(f_\phi(Z_t^{\text{hard}}, t), y(Z_0)\big). \tag{6}$$

Invalid endpoints remain traceable in the trajectory record but contribute no predictor gradient. A single Guide predictor is trained for each task, keeping the source of the sampling signal identifiable in the ablation.

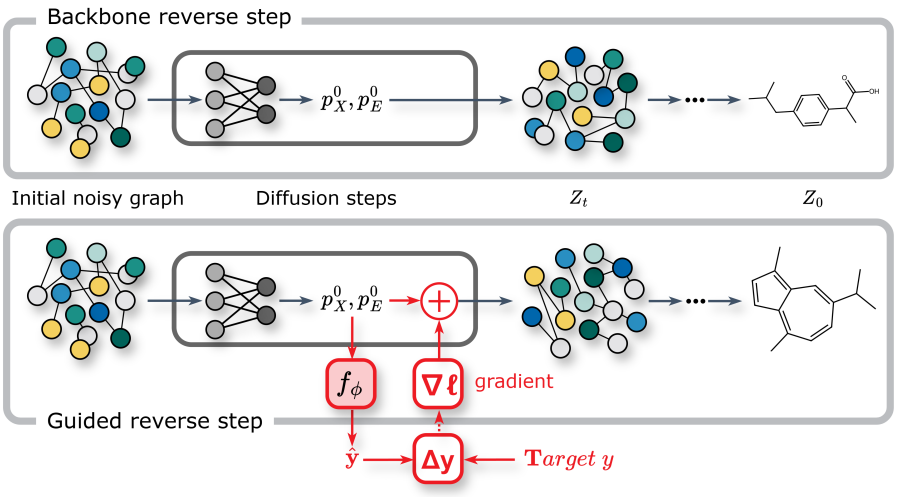

Figure 2: **Guide steers the reverse process toward the requested endpoint properties.** At reverse step t, the backbone produces the current clean-category predictions p_X^0 and p_E^0. Guide evaluates $\hat{\mathbf{y}}_t = f_\phi(p_X^0, p_E^0, t)$, compares it with target $\mathbf{y}$ through $\Delta \mathbf{y}_t$, and uses the resulting $\nabla_{p_X^0, p_E^0} \ell_t$ to modify the sampling distributions before the next hard state is drawn. The molecular structures are illustrative.

3.4 Guide as an active reverse-process controller

Endpoint prediction becomes guidance by evaluating the same network on the clean-category predictions p_X^0, p_E^0 and comparing the predicted endpoint with the requested target $\mathbf{y}$. We write

$$\hat{\mathbf{y}}_t = f_\phi(p_X^0, p_E^0, t), \qquad \Delta \mathbf{y}_t = \hat{\mathbf{y}}_t - \mathbf{y}. \tag{7}$$

For continuous properties and a requested binary class, respectively, the guidance losses are

$$\ell_t^{\text{cont}} = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} |\Delta y_{t,j}|, \tag{8}$$

$$\ell_t^{\text{cat}} = \text{BCEWithLogits}(\hat{y}_t, y). \tag{9}$$

We differentiate ℓ_t with respect to both p_X^0 and p_E^0 and perturb the sampling logits in the negative-loss direction using $\nabla_{p_X^0, p_E^0} \ell_t$. The update is restricted to active nodes, edge gradients are symmetrized, and the gradient norm is clipped before applying the task-specific guidance strength. Multiplication by t/T emphasizes earlier uncertain states and smoothly reduces the intervention near the endpoint. The ordinary backbone transition is then sampled from the guided distributions. Consequently, Guide acts throughout the reverse process: f_ϕ supplies a prospective endpoint estimate, and $\nabla \ell_t$ changes which graph the trajectory will generate.

3.5 Repair: chemistry-constrained edge resampling

Property guidance does not guarantee that a hard edge proposal is chemically coherent. At selected late timesteps, Repair summarizes a proposed graph with

$$\boldsymbol{\nu}(G) = \big(a(G), o(G), e(G), k(G)\big), \tag{10}$$

where a counts aromatic bonds outside valid cycles, o counts overvalent atoms, e is total valence excess, and k counts connected components beyond one. Candidate actions replace an illegal aromatic bond, lower or remove a bond incident to an overvalent atom, or connect components when endpoint capacity permits. Removing a bridge is forbidden because it would introduce a new disconnection.

Repair does not invent an independent molecular edit. For every candidate edge class, it reads the probability already assigned by the backbone posterior. Candidates are ordered by posterior support and violation reduction. An edit $G \to G'$ is committed only when

$$\boldsymbol{\nu}(G') \preceq \boldsymbol{\nu}(G) \quad \text{and} \quad \boldsymbol{\nu}(G') \neq \boldsymbol{\nu}(G), \tag{11}$$

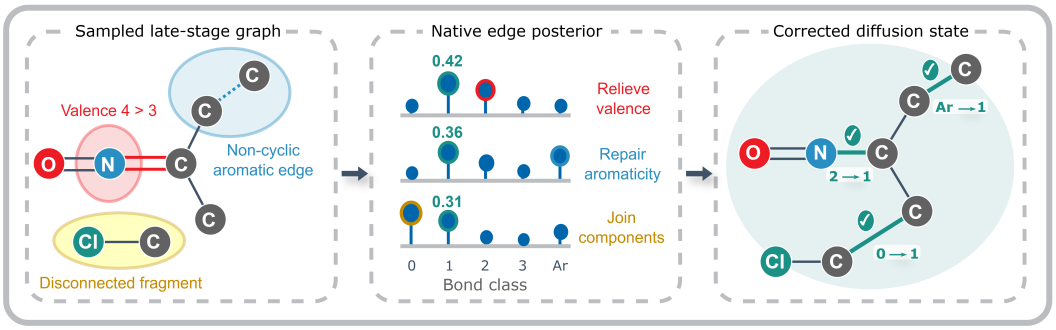

Figure 3: **Repair performs chemistry-constrained edge resampling inside the reverse chain.** A sampled late-stage graph is checked for valence, aromaticity, and connectivity violations. Edge labels 0/1/2/3/Ar denote no, single, double, triple, and aromatic bonds; posterior numbers are illustrative rather than measured values. Repair ranks alternative bond classes by the backbone's native edge posterior, accepts only violation-reducing edits, and returns the repaired state to the diffusion chain without changing atoms or graph size; it is not post-generation cleanup.

where the comparison is component-wise. Accepted edges remain symmetric, diagonal entries remain no-bond, and padding, node types, and node count are unchanged. The corrected hard graph is returned to the next reverse step rather than sanitized after generation.

Backbone-specific instantiation. PGR uses the same three control roles with both backbones, while respecting their native state spaces. For both MELD and Graph DiT, Guide is trained on one-hot encodings of sampled hard reverse states and is evaluated on probability-valued clean-category predictions during generation. Graph DiT exposes categorical node and edge posteriors directly; MELD excludes and renormalizes the mask edge class before Repair, leaves unresolved masks untouched, and writes accepted chemical edges back to its masked state. The continuous instantiation uses the L1 guidance loss over 200 reverse steps, whereas the categorical instantiation uses binary cross-entropy over 500 steps; Repair maps the same relative late-step locations to each horizon. These experiments establish compatibility with two backbone settings, not a controlled comparison between the backbones because task family and backbone change together.

3.6 Complete sampling procedure and cost

Given $\mathbf{c}$, PGR executes the following sequence: (1) Plan selects $\hat{N}$ and fixes the node mask; (2) the backbone predicts p_X^0, p_E^0 at every reverse step; (3) Guide computes $\hat{\mathbf{y}}_t$, $\Delta\mathbf{y}_t$, and ℓ_t, then uses $\nabla_{p_X^0, p_E^0} \ell_t$ to update those distributions before the ordinary hard transition; and (4) at scheduled late steps, Repair accepts a bounded number of posterior-supported edge replacements before the state continues through the chain. Plan adds one small predictor evaluation per molecule. Guide adds one predictor forward and backward pass per diffusion step. Repair is restricted to a few late steps and a bounded edit budget. None of the three modules changes the backbone parameters, and Repair never adds, deletes, or relabels nodes.

4 Experimental Setup

Research questions. We study whether PGR improves continuous and categorical conditional generation (**RQ1**), how Plan, Guide, and Repair contribute to alignment and raw validity (**RQ2**), and how guidance supervision, strength, and Repair affect control before evaluation-time correction (**RQ3**). We address these questions through benchmark comparisons, cumulative ablations, validation-only guidance analysis, and paired Guide-to-Repair transitions.

Tasks and backbones. We follow the conditional-generation protocols of Graph DiT (Liu et al., 2024) and MELD (Seo et al., 2026). The continuous task uses frozen MELD to jointly condition

Table 1: Property-conditioned polymer generation under the MELD evaluation protocol (Seo et al., 2026) (10,000 samples). Validity and Diversity are percentages, parentheses show raw validity, and bold marks the best displayed value.

	General Quality		Distribution Learning			Condition Control				
Method	Validity ↑	Coverage ↑	Diversity ↑	Similarity ↑	FCD ↓	Synth. ↓	O_2 ↓	N_2 ↓	CO_2 ↓	Avg. ↓
GraphGA	**100.00** (N/A)	**11/11**	88.28	0.927	9.19	1.3307	1.9840	2.2900	1.9489	1.888
MARS	**100.00** (N/A)	**11/11**	83.75	0.928	7.56	1.1658	1.5761	1.8327	1.6074	1.546
LSTM-HC	99.10 (N/A)	10/11	89.18	0.794	18.16	1.4251	1.1003	1.2365	1.0772	1.210
JTVAE-BO	**100.00** (N/A)	10/11	73.66	0.729	23.59	1.0714	1.0781	1.2352	1.0978	1.121
GDSS	92.05 (90.76)	9/11	75.10	0.000	34.26	1.3701	1.0271	1.0820	1.0683	1.137
MOOD	98.66 (92.05)	**11/11**	83.49	0.023	39.40	1.4019	1.4961	1.7603	1.4748	1.533
DiGress v2	98.12 (30.57)	**11/11**	**91.05**	0.278	21.73	2.7507	1.7130	2.0632	1.6648	2.048
Graph DiT	82.45 (84.37)	**11/11**	87.12	0.960	6.64	1.2973	0.7440	0.8857	0.7550	0.921
MELD	99.10 (96.51)	**11/11**	85.91	**0.974**	**5.93**	1.1398	0.6433	0.7596	0.6496	0.798
PGR	99.94 (**99.91**)	10/11	83.60	0.965	7.55	**0.6130**	**0.3962**	**0.4948**	**0.4824**	**0.497**

polymer generation on synthesizability and O_2/N_2/CO_2 permeability. The categorical tasks use frozen Graph DiT to pair synthesizability with BACE, BBBP, or HIV labels from MoleculeNet (Wu et al., 2018). Each task evaluates 10,000 generated graphs.

Comparisons and ablations. We compare PGR with representative molecular-optimization and diffusion baselines under their respective benchmark protocols. For the ten displayed categorical methods, Avg. Rank averages ascending synthesizability-error rank and descending property-accuracy rank, with average ranks for ties. Ablations cumulatively add Plan, Guide, and Repair while retaining the same backbone, conditions, and evaluation implementation within each task.

Guidance-strength sensitivity analysis. Strength is selected exclusively from validation generations using task-specific objectives; validity and Test results are excluded. Appendix B gives the grids, sample counts, and selection protocols.

Metrics. Corrected and raw validity are measured with and without benchmark rule-based conversion, respectively. Coverage measures represented categories; Diversity measures within-set variation; Similarity measures fragment proximity to the reference set; and FCD measures ChemNet distributional distance (Preuer et al., 2018). Continuous control uses synthesizability and O_2/N_2/CO_2 MAEs and their average, whereas categorical control uses synthesizability MAE and Property Accuracy. Experiments use WSL2 Linux, one 32-GB NVIDIA GeForce RTX 5090, and a 24-core Intel Core Ultra 9 285K; software details appear in the appendix.

5 Results

5.1 Continuous conditional generation

Table 1 demonstrates that PGR substantially strengthens continuous conditional control. It reduces Avg. MAE to 0.497, compared with 0.798 for MELD and 0.921 for Graph DiT, while achieving 99.91% raw validity. The close agreement between raw and corrected validity further shows that these gains are already present in the directly sampled graphs rather than being introduced by evaluation-time rule correction. Together, these results establish PGR as an effective approach for combining accurate multi-property control with near-perfect raw graph validity.

5.2 Categorical conditional generation

Table 2 shows that PGR's effectiveness extends consistently across all three categorical tasks. It achieves Property Accuracies of 0.9716, 0.9665, and 0.9912 on BACE, BBBP, and HIV, together with raw validities of 0.9248, 0.9549, and 0.7745. PGR also lowers Synth. MAE on every task and ranks

Table 2: Conditional generation of 10,000 molecules for BACE, BBBP, and HIV under the Graph DiT evaluation protocol (Liu et al., 2024). Parentheses show raw validity. Average rank is computed over the displayed methods from ascending Synth. MAE and descending Property Accuracy, with average ranks for ties; bold marks the best displayed value.

		General Quality		Distribution Learning			Condition Control		
Task	Method	Validity ↑	Coverage ↑	Diversity ↑	Similarity ↑	FCD ↓	Synth. ↓	Prop. Acc. ↑	Avg. Rank ↓
BACE	Graph GA	**1.0000** (N.A.)	**8/8**	0.8585	**0.9805**	7.4104	0.9633	0.4690	6.5000
	MARS	**1.0000** (N.A.)	**8/8**	0.8338	0.8827	**6.7923**	1.0123	0.5184	5.0000
	LSTM-HC	0.9972 (N.A.)	**8/8**	0.8146	0.7982	17.5585	0.9207	0.5816	3.0000
	JTVAE-BO	**1.0000** (N.A.)	6/8	0.6682	0.7281	30.4696	0.9923	0.4628	7.5000
	DiGress	0.3511 (0.2858)	**8/8**	0.8862	0.6942	24.6560	2.0681	0.5061	8.0000
	DiGress v2	0.3546 (0.2680)	**8/8**	0.8812	0.7027	25.3270	2.3365	0.5113	7.5000
	GDSS	0.2879 (0.2589)	4/8	0.8756	0.2708	46.7539	1.6422	0.5036	7.5000
	MOOD	0.9947 (0.4502)	**8/8**	**0.8902**	0.2587	44.2394	1.8853	0.5062	7.0000
	Graph DiT	0.8674 (0.8495)	**8/8**	0.8238	0.8752	7.0456	0.3998	0.9135	2.0000
	PGR	0.9351 (**0.9248**)	**8/8**	0.8116	0.8982	7.4785	**0.3100**	**0.9716**	**1.0000**
BBBP	Graph GA	**1.0000** (N.A.)	**9/9**	0.8950	**0.9509**	**10.1659**	1.2082	0.3015	7.5000
	MARS	**1.0000** (N.A.)	8/9	0.8637	0.7696	10.9791	1.2250	0.5189	6.0000
	LSTM-HC	0.9990 (N.A.)	8/9	0.8883	0.8932	16.3904	0.9969	0.5590	4.0000
	JTVAE-BO	**1.0000** (N.A.)	5/9	0.7458	0.5821	33.5746	1.1619	0.4958	6.0000
	DiGress	0.6960 (0.4871)	**9/9**	0.9098	0.6805	18.6921	2.3658	0.6536	6.5000
	DiGress v2	0.6892 (0.4100)	**9/9**	0.9107	0.6336	19.4498	2.2694	0.6531	6.5000
	GDSS	0.6218 (0.5919)	3/9	0.8415	0.2672	39.9440	1.3788	0.5037	7.0000
	MOOD	0.8008 (0.5789)	**9/9**	**0.9273**	0.1715	34.2506	2.0284	0.4903	8.5000
	Graph DiT	0.8468 (0.8505)	**9/9**	0.8856	0.9329	11.8519	0.3551	0.9417	2.0000
	PGR	0.9643 (**0.9549**)	**9/9**	0.8790	0.8781	14.3403	**0.2375**	**0.9665**	**1.0000**
HIV	Graph GA	**1.0000** (N.A.)	28/29	0.8993	**0.9661**	**4.4418**	0.9839	0.6035	5.0000
	MARS	**1.0000** (N.A.)	26/29	0.8764	0.6517	7.2893	0.9691	0.6455	4.0000
	LSTM-HC	0.9994 (N.A.)	13/29	0.9091	0.9145	7.4659	0.9480	0.6736	3.0000
	JTVAE-BO	**1.0000** (N.A.)	3/29	0.8055	0.4173	41.9771	1.2359	0.4850	7.5000
	DiGress	0.4377 (0.3643)	22/29	0.9194	0.8562	13.0409	1.9216	0.5335	7.5000
	DiGress v2	0.5050 (0.4242)	24/29	0.9193	0.8476	13.3997	1.5934	0.5331	7.5000
	GDSS	0.6926 (0.6757)	4/29	0.7817	0.1032	45.3416	1.2515	0.4830	8.5000
	MOOD	0.2875 (0.2173)	**29/29**	**0.9280**	0.1361	32.3523	2.3144	0.5106	9.0000
	Graph DiT	0.7660 (0.7415)	28/29	0.8974	0.9575	6.0216	0.3086	0.9777	2.0000
	PGR	0.8014 (**0.7745**)	28/29	0.8931	0.9228	9.5756	**0.2235**	**0.9912**	**1.0000**

first by the two-metric average in all three comparisons. These consistent improvements demonstrate that PGR provides robust conditional control across distinct biomedical property objectives.

5.3 Ablating Plan, Guide, and Repair

Figure 4 demonstrates the cumulative contributions of the three controls, with complete results in Appendix Table S4. On the continuous task, Plan reduces Avg. MAE from 0.7981 to 0.5692, and Guide further lowers it to 0.5050, compared with 0.5824 for forward-noising supervision at the same strength (Appendix Table S3). Guide also improves Property Accuracy on BACE, BBBP, and HIV, confirming the benefit of reverse-trajectory supervision.

Repair raises continuous raw validity from 0.9825 to 0.9991 while reducing Avg. MAE from 0.5050 to 0.497. It also increases raw validity and Property Accuracy on all three categorical tasks. Together, these results support the intended roles of Plan in size selection, Guide in endpoint control, and Repair in late structural correction without erasing alignment gains.

5.4 Repair acts before evaluation-time correction

Table 3 shows that raw-invalid→valid transitions outnumber the reverse direction on every task, including 172 versus 6 on $O_2/N_2/CO_2$. Together with the close corrected and raw validities, these paired counts show that Repair improves sampler outputs rather than relying on evaluation-time correction.

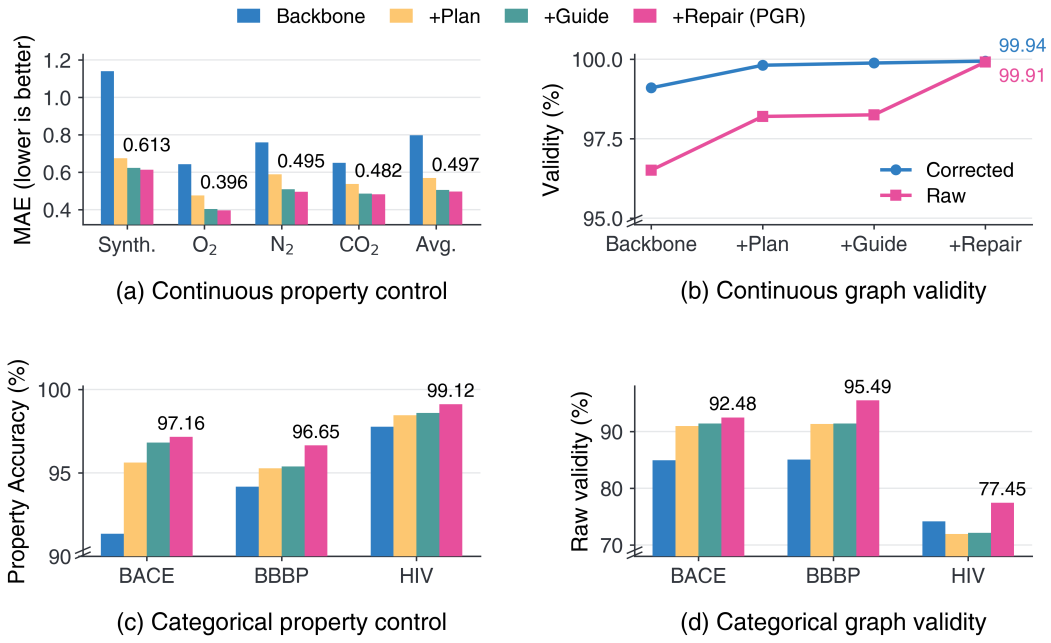

Figure 4: **Stage-wise ablation of Plan, Guide, and Repair.** (a) Synthesizability, O_2, N_2, CO_2, and average property MAEs across Backbone, Plan, Guide, and Repair on the continuous task. (b) Corrected and raw validity along the same continuous-stage sequence. (c) Property Accuracy and (d) raw validity across the four stages on BACE, BBBP, and HIV.

Table 3: Paired Guide→Repair transitions for the same 10,000 conditions and Plan sizes.

Task	Changed SMILES	Invalid → valid	Valid → invalid	Wrong → correct	Correct → wrong
$O_2/N_2/CO_2$	1,015	172	6	—	—
BACE	2,955	420	312	524	490
BBBP	4,011	527	119	994	868
HIV	2,827	792	262	962	910

6 Analysis and Limitations

Why the stages are complementary. Plan acts before topology exists, Guide steers evolving reverse states, and Repair corrects resolved late bond proposals. The ablation supports this staged design without assigning molecule-level causality.

Limitations. PGR relies on surrogate predictors, adds sampling cost, and may require task- and backbone-specific validation scans. Repair covers the tested atom–bond representation; richer chemistry and geometry require tailored constraints. FCD, Similarity, and Diversity do not improve uniformly.

7 Conclusion

PGR separates size planning, trajectory-endpoint guidance, and chemistry-constrained edge repair. Across one continuous and three categorical benchmarks, it improves condition alignment while attaining strong raw validity, with ablations supporting each stage. Its scope remains property control and graph validity for the tested representations.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, bibliographic metadata checking, figure-code drafting, and LaTeX preparation. They were not used to generate experimental observations or change measured results. The authors reviewed all AI-assisted text, citations, figures, and source files and take responsibility for the final manuscript.

ETHICS STATEMENT

This work develops a computational sampling method for molecular design and does not involve human participants or private data. Generated molecules are hypotheses rather than validated compounds. Their downstream use requires independent chemical, safety, environmental, and application-specific assessment; the reported property predictors and validity checks do not establish experimental efficacy or safety.

REPRODUCIBILITY STATEMENT

The appendix specifies the algorithms, training-data separation, guidance-strength protocols, sampling settings, evaluation definitions, and implementation environment needed to interpret the experiments. Each reported evaluation contains 10,000 generated samples.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Fan Bao, Min Zhao, Zhongkai Hao, Peiyao Li, Chongxuan Li, and Jun Zhu. Equivariant energy-guided SDE for inverse molecular design. In *International Conference on Learning Representations*, 2023. OpenReview:r0otLtOwYW; arXiv:2209.15408.

J. Wesley Barnett, Connor R. Bilchak, Yiwen Wang, Brian C. Benicewicz, Laura A. Murdock, Tristan Bereau, and Sanat K. Kumar. Designing exceptional gas-separation polymer membranes using machine learning. *Science Advances*, 6(20):eaaz4301, 2020. doi: 10.1126/sciadv.aaz4301.

Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. Guacamol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3): 1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. arXiv:2205.14987.

Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. SCScore: Synthetic complexity learned from a reaction corpus. *Journal of Chemical Information and Modeling*, 58(2): 252–261, 2018. doi: 10.1021/acs.jcim.7b00622.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Yuanqi Du, Arian R. Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L. Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, 6:589–604, 2024. doi: 10.1038/s42256-024-00843-5.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, 2009. doi: 10.1186/1758-2946-1-8.

Tianfan Fu, Wenhao Gao, Cao Xiao, Jacob Yasonik, Connor W. Coley, and Jimeng Sun. Differentiable scaffolding tree for molecular optimization. In *International Conference on Learning Representations*, 2022. arXiv:2109.10469.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21342–21357, 2022. arXiv:2206.12411.

Niklas W. A. Gebauer, Michael Gastegger, Stefaan S. P. Hessmann, Klaus-Robert Müller, and Kristof T. Schütt. Inverse design of 3d molecular structures with conditional generative neural networks. *Nature Communications*, 13:973, 2022. doi: 10.1038/s41467-022-28526-y.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, 2021. arXiv:2102.05379.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887, 2022.

Jan H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2323–2332, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4849–4859, 2020.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10362–10383, 2022.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. arXiv:1312.6114.

Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18872–18892, 2023.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph diffusion transformers for multi-conditional molecular generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2401.13858.

Ruimin Ma and Tengfei Luo. PI1M: A benchmark database for polymer informatics. *Journal of Chemical Information and Modeling*, 60(10):4684–4690, 2020. doi: 10.1021/acs.jcim.0c00726.

Amina Mollaysa, Brooks Paige, and Alexandros Kalousis. Goal-directed generation of discrete structures with conditional generative models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21923–21933, 2020. arXiv:2010.02311.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021. arXiv:2102.09672.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4474–4484, 2020.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:565644, 2020. doi: 10.3389/fphar.2020.565644.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. doi: 10.1021/acs.jcim.8b00234.

Philipp Renz, Dries Van Rompaey, Jörg Kurt Wegner, Sepp Hochreiter, and Günter Klambauer. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, 32–33:55–63, 2019. doi: 10.1016/j.ddtec.2020.09.003.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, 2015.

Lloyd M. Robeson. The upper bound revisited. *Journal of Membrane Science*, 320(1–2):390–400, 2008. doi: 10.1016/j.memsci.2008.04.030.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4135.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018. doi: 10.1126/science.aat2663.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1): 120–131, 2018. doi: 10.1021/acscentsci.7b00512.

Hyunjin Seo, Taewon Kim, Sihyun Yu, and Sungsoo Ahn. Learning flexible forward trajectories for masked molecular diffusion. In *International Conference on Learning Representations*, 2026. arXiv:2505.16790.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: A flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2020. arXiv:2001.09382.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3277.

Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, volume 11139 of *Lecture Notes in Computer Science*, pp. 412–422, 2018. doi: 10.1007/978-3-030-01418-6_41.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. arXiv:2011.13456.

Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation. *arXiv preprint arXiv:2310.09267*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. arXiv:1706.03762.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Tomer Weiss, Eduardo Mayo Yanes, Sabyasachi Chakraborty, Luca Cosmo, Alex M. Bronstein, and Renana Gershoni-Poranne. Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882, 2023. doi: 10.1038/s43588-023-00532-0.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/C7SC02664A.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. MARS: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations*, 2021. arXiv:2103.10432.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, volume 31, 2018. arXiv:1806.02473.

Chengxi Zang and Fei Wang. MoFlow: An invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 617–626, 2020. doi: 10.1145/3394486.3403104.

A Appendix overview

This appendix provides additional methodological and experimental details for PGR. Appendix B expands the algorithms for Plan, Guide, and Repair and reports the validation-only guidance-strength selection and guidance-supervision comparison. Appendix C provides the complete numerical stage-wise ablation underlying the compact main-text visualization. Appendix D describes the tasks, evaluation definitions, sampling settings, and implementation environment.

B Detailed algorithms and guidance selection

B.1 Plan

For each task, a node-count predictor receives the same condition representation used by the generator. Counts absent from the corresponding training split are masked before normalization. The selected size is the maximum-probability supported count and remains fixed for the complete reverse process. Continuous conditions are standardized numerical values. In categorical tasks, continuous synthesizability descriptors remain numerical and the activity target remains categorical.

Algorithm S1: condition-aware molecular-size planning.

1. Fit $q_\omega(N \mid \mathbf{c})$ using only training conditions and their graph sizes.
2. Construct support $\mathcal{N}_{\text{train}}$ from node counts observed in the training split.
3. For a requested condition $\mathbf{c}^*$, choose
$$\hat{N}(\mathbf{c}^*) = \arg\max_{N \in \mathcal{N}_{\text{train}}} q_\omega(N \mid \mathbf{c}^*).$$
4. Instantiate a node mask with $\hat{N}$ active entries and retain it through all reverse steps.

B.2 Guide

Trajectory collection. The frozen backbone is sampled with Plan active and property gradients inactive. Only train-split conditions enter trajectory construction. Each of 12,000 trajectories contributes one sampled categorical reverse state and its final generated graph. The sampled node and edge categories are stored as one-hot predictor inputs; they are not reconstructed by taking an argmax of a soft posterior. Stored timesteps cover the reverse horizon uniformly: 60 states per step for the 200-step continuous backbone and a balanced allocation over 500 steps for the categorical backbone. A condition identifier keeps all records from a condition in the same fit, selection, or holdout partition.

Endpoint labels. The label belongs to the final generated graph, not to the input condition and not to a clean training molecule. Invalid endpoints remain in the trajectory data but are masked from predictor loss. For the continuous task, train-only random-forest teachers label the O_2, N_2, and CO_2 endpoint properties in log space; the guidance loss then uses the established raw-property normalization. For categorical tasks, train-only classifiers supply endpoint labels for BACE, BBBP, or HIV. This separation prevents test molecules from entering predictor training.

Predictor and optimization. The trajectory predictor uses four graph-attention layers, hidden dimension 384, eight attention heads, bfloat16 arithmetic, batch size 256, learning rate 6×10^{-4}, at most 300 epochs, patience 10, and seed 0. Conditions are split at the condition level into fit, selection, and holdout subsets. Continuous training minimizes normalized endpoint L1 over the three gas properties; categorical training minimizes endpoint binary cross-entropy. The predictor with the best selection loss is used for sampling. Predictor holdout metrics are diagnostic and are not substituted for generated-sample metrics.

Algorithm S2: trajectory-endpoint guidance.

1. Sample a planner-conditioned, guidance-free reverse trajectory and store one sampled categorical state Z_t^{hard} together with endpoint Z_0.

Table S1: Continuous-task 2K validation grid. The selection quantity is the equally weighted four-property Avg. MAE.

Strength	50	60	70	**80**	90	100	110	120
Avg. MAE	0.511753	0.499007	0.499566	**0.496097**	0.499616	0.504978	0.502005	0.508973
Strength	130	140	150	160	170	180	190	200
Avg. MAE	0.512395	0.512814	0.513076	0.510628	0.506648	0.508196	0.508586	0.506420
Strength	300	400	500	600	700	800	900	1000
Avg. MAE	0.507831	0.565831	0.550767	0.548435	0.554735	0.537161	0.549544	0.553213

Table S2: Categorical guidance-strength protocols. Each validation candidate contains 2,000 generations.

Task	Coarse grid	Local dense window	Best value	Selected strength
BACE	50–1000 / 50	850–1000 / 10	0.949133	940
BBBP	50–1000 / 50	10–200 / 10	0.925277	10
HIV	50–1000 / 50	550–750 / 10	0.965004	650

2. Compute $\mathbf{y}(Z_0)$ offline; set a mask to zero if Z_0 is invalid.
3. One-hot encode the sampled categories and train $f_\phi(Z_t^{\text{hard}}, t)$ to predict $\mathbf{y}(Z_0)$ with the corresponding continuous or categorical masked endpoint loss defined in the main paper.
4. During generation, compute $\hat{\mathbf{y}}_t = f_\phi(p_X^0, p_E^0, t)$, $\Delta\mathbf{y}_t = \hat{\mathbf{y}}_t - \mathbf{y}$, and the task-specific guidance loss ℓ_t.
5. Backpropagate $\nabla_{p_X^0, p_E^0}\ell_t$, mask padded nodes, symmetrize edge gradients, clip the gradient norm at 10, multiply by t/T, and update the sampling distributions before the ordinary reverse transition.

The continuous guidance strength is 80. The categorical strengths are 940 for BACE, 10 for BBBP, and 650 for HIV. Their validation-only selection protocols are described next.

B.3 Guidance-strength sensitivity and settings

Common protocol. The validation sensitivity analyses use 2,000 generations per candidate while holding the requested conditions and Plan node counts fixed within each task. Exact ties favor the smaller strength. Validity and Test outcomes do not enter the ranking objective. The continuous backbone uses 200 reverse steps and a generation batch size of 2048; the categorical backbone uses 500 reverse steps and a generation batch size of 512.

Continuous task. The selection quantity is

$$\mathcal{M}_{\text{sel}} = \frac{\text{MAE}_{\text{Synth}} + \text{MAE}_{\text{O}_2} + \text{MAE}_{\text{N}_2} + \text{MAE}_{\text{CO}_2}}{4}. \tag{12}$$

Lower is better. Strength 80 attains the lowest 2K validation Avg. MAE, 0.496097, and is fixed for Test evaluation. The complete continuous grid is reported in Table S1.

Categorical tasks. BACE, BBBP, and HIV independently maximize validation Property Accuracy. This selects 940, 10, and 650, respectively. Validity and Test outcomes do not enter the curves or ranking. Table S2 summarizes the candidate grids and selected strengths.

Figure S1 visualizes all validation candidates for the continuous and categorical tasks.

Guidance-supervision comparison. Table S3 compares forward-noising supervision, trajectory-endpoint supervision, and complete PGR on the continuous task. All rows use the same Plan predictions, requested conditions, and validation-selected guidance strength.

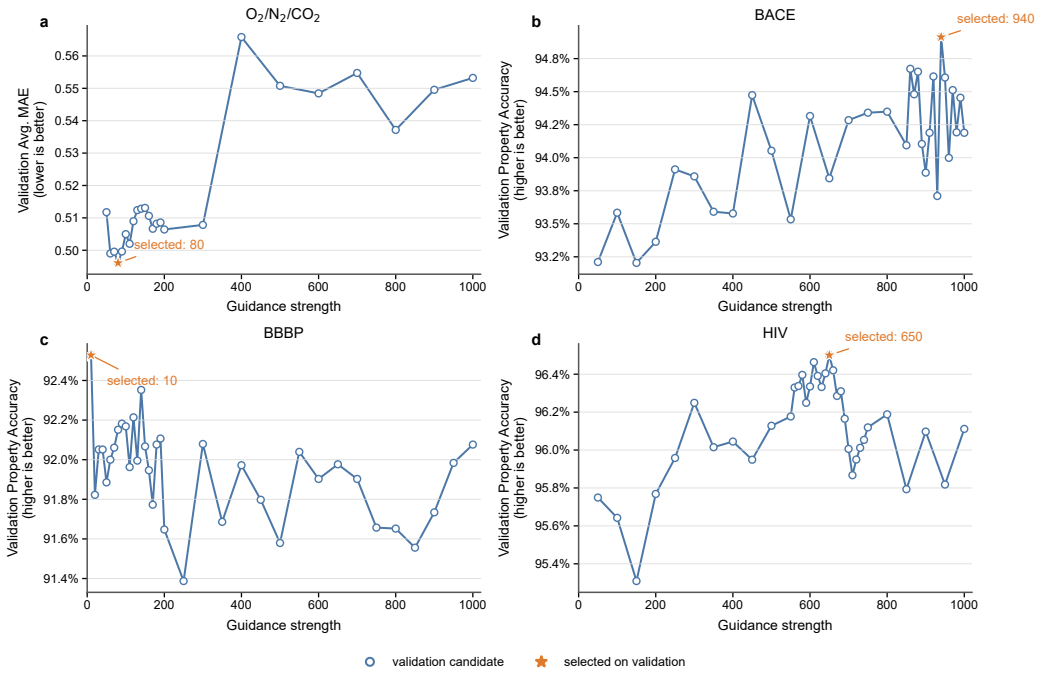

Figure S1: **Guidance-strength sensitivity analysis on validation generations.** The continuous panel contains 24 strengths and reports Avg. MAE (lower is better). The BACE, BBBP, and HIV panels contain 32, 36, and 36 unique strengths and report Property Accuracy (higher is better). All candidates use hollow markers, and orange stars mark the validation-selected strengths 80, 940, 10, and 650. Every point contains 2,000 validation generations; Validity and Test outcomes are excluded from selection. No error bars are shown because each candidate contributes one aggregate estimate from 2,000 validation generations.

Table S3: Continuous-task comparison of property guidance supervision and Repair. All rows use the same Plan predictions, requested conditions, and validation-selected guidance strength. Values in parentheses are raw validity.

Method	Validity ↑	Coverage ↑	Diversity ↑	Similarity ↑	FCD ↓	Synth. ↓	O_2 ↓	N_2 ↓	CO_2 ↓	Avg. ↓
Forward-noising Guide	0.9985 (0.9749)	10/11	**0.8386**	0.9573	8.1655	0.6170	0.5125	0.6195	0.5804	0.5824
Trajectory-endpoint Guide	0.9988 (0.9825)	10/11	0.8370	0.9639	**7.5109**	0.6227	0.4036	0.5086	0.4852	0.5050
PGR	**0.9994 (0.9991)**	10/11	0.8360	**0.9651**	7.5540	**0.6130**	**0.3962**	**0.4948**	**0.4824**	**0.4966**

B.4 Repair

Chemical capacities. An atom-type capacity table is fixed before generation. Each capacity is the smaller of the maximum valence observed in the task training split and the maximum allowed valence from the chemistry toolkit. If the toolkit does not support an element, the observed training maximum is used; a type is unconstrained only when both are unknown. The toolkit is used offline for capacity construction and final evaluation, not during generation.

Candidate construction and acceptance. At an active timestep, Repair receives a hard undirected graph and the backbone's edge posterior. It constructs candidates that (i) replace illegal aromatic bonds, (ii) lower or remove bonds adjacent to an overvalent node, or (iii) connect components when endpoint capacities allow. A bridge is protected from deletion. Candidates are ranked by posterior probability and then by improvement in the violation tuple. The first candidate that improves at least one tuple component without worsening another is committed, after which candidates are regenerated. The procedure stops when no acceptable candidate remains or the edit budget is reached.

Algorithm S3: chemistry-constrained edge repair.

1. At a scheduled late step, compute $\boldsymbol{\nu}(G) = (a, o, e, k)$.
2. Enumerate symmetric edge replacements permitted by the node mask and capacity table.
3. Remove bridge-deleting candidates and order the remainder by posterior support, followed by violation improvement.
4. Commit an edit only if $\boldsymbol{\nu}(G') \preceq \boldsymbol{\nu}(G)$ and at least one component decreases.
5. Repeat up to 16 accepted edits; pass the resulting graph to the next reverse step.

For the 500-step categorical backbone, active steps are $[50, 40, 30, 20, 10]$. For the 200-step continuous backbone, the same relative locations map to $[20, 16, 12, 8, 4]$. Repair does not consume CPU or accelerator random-number state, change node types or counts, or modify entries outside the active mask.

C Complete stage-wise ablation results

Figure 4 provides a compact view of the cumulative effects of Plan, Guide, and Repair. Table S4 reports the complete numerical results for the same stage-wise comparisons, including the distributional metrics omitted from the visualization. Within each task, the Plan, Guide, and Repair rows share the same requested conditions and Plan node predictions. Together, the results show how condition-aware size selection, trajectory-endpoint guidance, and late structural repair contribute cumulatively to property control and raw graph validity.

Table S4: Complete numerical results for the stage-wise PGR ablation summarized in Figure 4. Plan, Guide, and Repair form a strictly paired chain. Corrected validity includes benchmark conversion, whereas raw validity does not. The shaded rows denote complete PGR, and bold marks the best value within each task.

(a) O_2/N_2/CO_2

Stage	Corr. val. ↑	Raw val. ↑	Diversity ↑	Similarity ↑	FCD ↓	Synth. ↓	O_2 ↓	N_2 ↓	CO_2 ↓	Avg. ↓
Backbone	0.9910	0.9651	**0.8591**	**0.9740**	**5.9300**	1.1398	0.6433	0.7596	0.6496	0.7981
+ Plan	0.9981	0.9820	0.8430	0.9613	7.7027	0.6748	0.4760	0.5893	0.5369	0.5692
+ Guide	0.9988	0.9825	0.8370	0.9639	7.5109	0.6227	0.4036	0.5086	0.4852	0.5050
+ Repair (PGR)	**0.9994**	**0.9991**	0.8360	0.9651	7.5540	**0.6130**	**0.3962**	**0.4948**	**0.4824**	**0.4966**

(b) BACE, BBBP, and HIV

Task	Stage	Corr. val. ↑	Raw val. ↑	Diversity ↑	Similarity ↑	FCD ↓	Synth. ↓	Prop. Acc. ↑
BACE	Backbone	0.8674	0.8495	**0.8238**	0.8752	**7.0456**	0.3998	0.9135
	+ Plan	0.9220	0.9098	0.8094	0.8955	7.4785	**0.3074**	0.9562
	+ Guide	0.9259	0.9140	0.8105	**0.8984**	7.2079	0.3111	0.9682
	+ Repair (PGR)	**0.9351**	**0.9248**	0.8116	0.8982	7.4785	0.3100	**0.9716**
BBBP	Backbone	0.8468	0.8505	**0.8856**	**0.9329**	**11.8519**	0.3551	0.9417
	+ Plan	0.9156	0.9134	0.8778	0.8787	14.0064	0.2762	0.9527
	+ Guide	0.9162	0.9141	0.8780	0.8750	14.0296	0.2782	0.9539
	+ Repair (PGR)	**0.9643**	**0.9549**	0.8790	0.8781	14.3403	**0.2375**	**0.9665**
HIV	Backbone	0.7660	0.7415	**0.8974**	**0.9575**	**6.0216**	0.3086	0.9777
	+ Plan	0.7480	0.7192	0.8924	0.9269	9.1530	0.2588	0.9846
	+ Guide	0.7491	0.7215	0.8927	0.9266	9.1317	0.2544	0.9860
	+ Repair (PGR)	**0.8014**	**0.7745**	0.8931	0.9228	9.5756	**0.2235**	**0.9912**

D Tasks, evaluation, and implementation environment

The continuous experiment jointly conditions polymer graphs on synthesizability and three gas-permeability targets, following the MELD protocol (Seo et al., 2026). The categorical experiments follow Graph DiT (Liu et al., 2024) and condition small-molecule generation on synthesizability and a BACE, BBBP, or HIV label from the MoleculeNet setting (Wu et al., 2018). Each reported

evaluation contains 10,000 generated samples. Plan, Guide, and Repair within a task share the same backbone checkpoint, evaluation implementation, requested conditions, and Plan node predictions.

Corrected validity is benchmark validity after rule-based checking; raw validity omits that correction. Coverage, Diversity, Similarity, FCD, synthesizability MAE, gas-property MAE, and Property Accuracy follow the cited benchmark protocols. The categorical Avg. Rank is computed over the ten methods displayed in the main table as the mean of ascending synthesizability-error rank and descending Property-Accuracy rank, using average ranks for ties.

Experiments run under Linux 6.6.114.1 in WSL2 on one 32-GB NVIDIA GeForce RTX 5090 and an Intel Core Ultra 9 285K CPU with 24 cores. The software environment uses Python 3.9.16, PyTorch 2.7.0+cu128, CUDA runtime 12.8, RDKit 2023.09.4, and PyTorch Geometric 2.3.0.